



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09319635 A**(43) Date of publication of application: **12.12.97**

(51) Int. Cl.

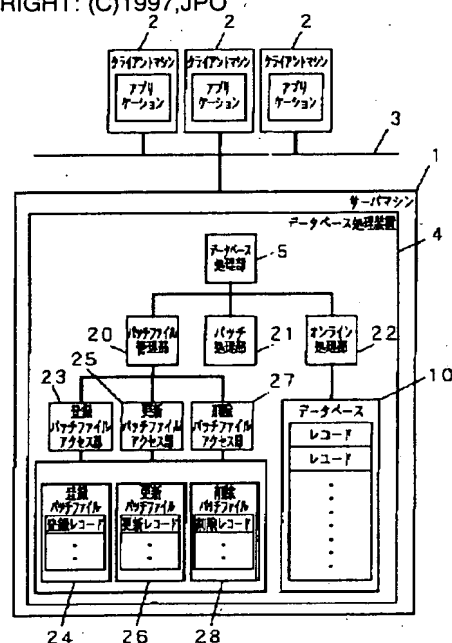
G06F 12/00**G06F 17/30**(21) Application number: **08136233**(22) Date of filing: **30.05.96**(71) Applicant: **MATSUSHITA GRAPHIC COMMUN
SYST INC**(72) Inventor: **WATANABE SHINICHI**(54) **DATABASE UPDATING METHOD**

COPYRIGHT: (C)1997,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To accelerate batch processing or update processing itself of records in a database by performing margin processing to the received update records having the same ID, and updating the records registered in the database while using the records after the margin processing.

SOLUTION: A batch file managing part 20 is activated and it is checked whether the record of the same ID as a record becoming the object of update processing is defined as the object of new registration or not (exists in an update file 26 or not). When the record of the same ID exists in the update batch file 26, the batch file managing part 20 performs the margin processing for the unit of a field to the update record and the update record existent in the update batch file 26 through an update batch file access part 25, generates the new update record, deletes the existent update record and replaces the generated new update record with the deleted update record.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-319635

(43) 公開日 平成9年(1997)12月12日

(51) Int.Cl.⁸

G 0 6 F 12/00
17/30

識別記号

5 1 8

庁内整理番号

F I

G 0 6 F 12/00
15/401

技術表示箇所

5 1 8 M
3 4 0 A

審査請求 未請求 請求項の数10 O L (全 15 頁)

(21) 出願番号

特願平8-136233

(22) 出願日

平成8年(1996)5月30日

(71) 出願人 000187736

松下電送株式会社

東京都目黒区下目黒2丁目3番8号

(72) 発明者 渡辺 紳一

東京都目黒区下目黒2丁目3番8号 松下
電送株式会社内

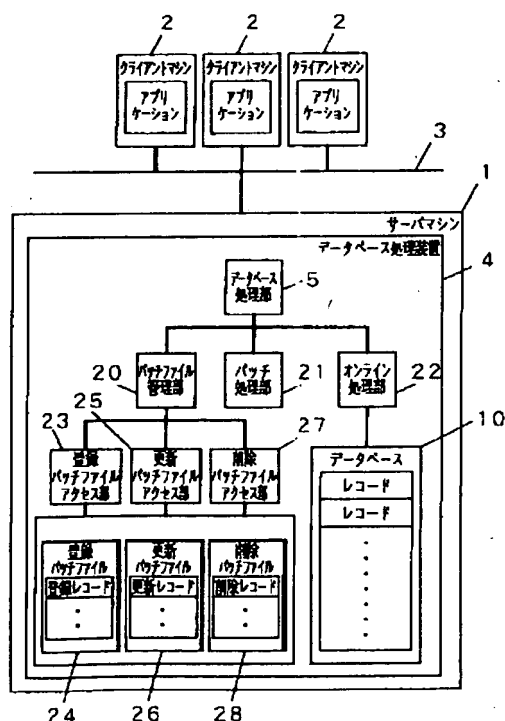
(74) 代理人 弁理士 滝本 智之 (外1名)

(54) 【発明の名称】 データベース更新方法

(57) 【要約】

【課題】 バッチ処理の対象となるレコード数の削減、バッチ処理の対象となるレコードデータ量の削減を図ることにより、データベースのレコードの更新を迅速に行なうこと。

【解決手段】 データベースに記録されたレコードに対する更新指示と更新レコードとを順次受付け、同一IDを有する既登録の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新する構成とした。



【特許請求の範囲】

【請求項1】 データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、同一IDを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新することを特徴とするデータベース更新方法。

【請求項2】 データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、その更新レコードと同一IDを有する受付済の更新レコードとをマージ処理して生成した単一の新更新レコードを更新レコードとして保持した後、データベースのレコードをこの新更新レコードにより更新することを特徴とするデータベース更新方法。

【請求項3】 データベースに登録されたレコードに対する更新指示と更新レコードとを順次受付け、順次受付けた同一IDを有する複数の更新レコードをマージ処理してID毎に単一のファイルとした上で、前記データベースのレコードの更新処理を実行することを特徴とするデータベース更新方法。

【請求項4】 マージ処理により生成される新更新レコードを順次保存する一方、マージ処理の対象としたその更新レコードと同一IDを有する受付済の更新レコードを順次消去することを特徴とする請求項1乃至請求項3記載のデータベース更新方法。

【請求項5】 最終的に削除対象となるレコードと同一レコードに対する登録指示又は更新指示がある場合には、その登録指示又は更新指示を無効にすることを特徴とする請求項1乃至請求項4記載のデータベース更新方法。

【請求項6】 データベースのレコードの更新処理実行前の更新レコードを登録処理バッチファイルと更新処理バッチファイルと削除処理バッチファイルとに分割して管理することを特徴とする請求項1乃至請求項4記載のデータベース更新方法。

【請求項7】 登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一IDのレコードをデータベースのレコード更新以前に消去することを特徴とする請求項6記載のデータベース更新方法。

【請求項8】 レコードの更新指示を時刻管理し、同一レコードに対して複数の更新指示がある場合は、新しい時刻の更新指示のみを実行することを特徴とする請求項1乃至請求項7記載のデータベース更新方法。

【請求項9】 データベースに対する検索の実行の後、その検索結果レコードと更新処理待ちの更新レコードとをマージ処理することを特徴とする請求項1乃至請求項8記載のデータベース更新方法。

【請求項10】 データベースに登録されたレコードの更新指示受付けの際にレコードの更新に要する時間を計算し、その時間が一定値以下である場合には、更新処理

をリアルタイムで実行し、その時間が一定値以上である場合には、更新処理をバッチ処理で実行することを特徴とする請求項1乃至請求項9記載のデータベース更新方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ネットワークによって接続されたクライアント、サーバ型データベースシステムにおけるレコードの登録、更新、削除等の更新処理をバッチ処理で実行するデータベース更新方法に関する。

【0002】

【従来の技術】 近年、ネットワークの普及に伴いクライアント、サーバ型データベースシステムが増加している。このようなLAN等のネットワークによって接続されたクライアント、サーバ型データベースシステムにおいて、大量データを処理する場合などレコードに対する登録、更新、削除処理が遅い場合には、検索処理はオンラインでリアルタイムに処理し、登録、更新、削除のデータの更新処理は登録、更新、削除レコードをオンラインでバッチファイルに保持し、オフライン時にバッチ処理で処理する形態が取られている。

【0003】 以下に、このような従来のクライアント、サーバ型データベースシステムについて図面を用いて説明する。図14は、従来のクライアント、サーバ型データベースシステムを示すものである。

【0004】 図14において、1は、WS、PC、ホストマシン等のサーバマシン、2は、検索、登録、更新、削除処理を要求するアプリケーションを搭載した、WS、PC等のクライアントマシン、であり、サーバマシン1とクライアントマシン2とは、LAN、WAN等のネットワーク3で接続されている。4は、サーバマシン1に実装されたデータベース処理装置、5は、クライアントからの処理要求を受取り、オンライン時にクライアントからの処理要求を受取り、検索、登録、更新、削除処理を実行するデータベース処理部、6は、クライアントマシン2からのレコードの登録、更新、削除の要求に応じ、オンライン時にリアルタイムにバッチファイル7を作成するバッチファイル管理部、8は、オフライン時にバッチファイルのレコードのバッチ処理を実行するバッチ処理部、9は、オンライン時にリアルタイムにデータベース10を構成するレコードの検索処理等を実行するオンライン処理部である。

【0005】 データベース10に登録されているレコードは、例えば、図15に示す形態であり、1レコードは、レコードID11と、そのデータ内容が、それぞれ、"AAA"、"BBB"、"CCC"である3つのフィールド12とから構成されている。また、図17は、バッチファイル7内の1レコードを示したものであり、レコードID11とフィールド12の他に登録、更

10

20

30

40

50

新、削除の処理の別を示す処理種別13を示すフィールドを有している。このバッチファイル7内に登録されている各レコードは、順次時系列に受け付けられる。

【0006】以上のように構成されたクライアント、サーバ型データベースシステムの動作概要を、図16のフロー図に沿って説明する。

【0007】まず、クライアントマシン2のアプリケーションは、サーバマシン1のデータベース処理装置4に対し、レコードの検索処理、あるいは、登録処理、あるいは、更新処理、あるいは、削除処理を依頼し(ST1)、データベース処理部5が、その要求を受け付け(ST2)、各処理種別に応じて次の処理を行う。

【0008】検索処理の場合は、オンライン処理部9を起動し(ST3)、データベース10から検索条件に合致したレコードを検索し、検索結果レコードをクライアントマシンへ返す(ST4)。

【0009】登録処理、更新処理、削除処理の場合には、バッチファイル管理部6を起動し(ST5)、オンラインでバッチファイル7へ登録レコード、更新レコード、削除レコードを図17のように時系列に順次追加し(ST6)、オフライン時に、バッチ処理部8で、データベース10へのレコードの登録処理、更新処理、削除処理を実行する(ST7)。

【0010】

【発明が解決しようとする課題】しかしながら、上記の従来のクライアント、サーバ型データベースシステムの構成では、時系列にバッチファイルが作成されるため、オフライン時のバッチ処理の際に、同一レコードに対する更新、削除処理がその都度実行されるので、これらの処理が大量に発生する場合には、膨大に時間を要するという問題がある。

【0011】また、バッチ処理そのものの問題として、バッチファイルへの登録を完了しても、データベースのレコードの実体が更新されなければ、検索対象にはならないため検索結果に反映されないという問題がある。

【0012】また、レコードのデータ量とサーバマシンの負荷等との関係で、登録、更新、削除等の処理がリアルタイムで実行できる程に短時間で済む場合でも、バッチ処理対象となり、実際の更新処理が遅れ、更新レコードの増大に応じて、オフライン時のバッチ処理の負荷も増大するという問題がある。

【0013】本発明は、これらの課題に鑑みて為されたものであり、バッチ処理の対象となるレコード数の削減、バッチ処理の対象となるレコードデータ量の削減を図ることにより、データベースのレコードの更新を迅速に行なうとともに、バッチ処理自体の負荷を軽減することにより、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現するデータベース更新方法を提供することを目的とする。

【0014】

【課題を解決するための手段】これらの目的を達成するために、本発明のデータベース更新方法は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受け付け、同一IDを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新するという構成をとる。

【0015】これにより、データベースのレコードの更新を迅速に行うとともに、バッチ処理自体の負荷を軽減することにより、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現することができる。

【0016】

【発明の実施の形態】本発明の請求項1記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受け付け、同一IDを有する受付済の更新レコードに対してマージ処理を行ない、データベースのレコードをそのマージ処理後のレコードを用いて更新するようにしたものである。

【0017】また、請求項2記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受け付け、その更新レコードと同一IDを有する受付済の更新レコードとをマージ処理して生成した単一の新更新レコードを更新レコードとして保持した後、データベースのレコードをこの新更新レコードにより更新するようにしたものである。

【0018】また、請求項3記載の発明は、データベースに登録されたレコードに対する更新指示と更新レコードとを順次受け付け、順次受け付けた同一IDを有する複数の更新レコードをマージ処理してID毎に単一のファイルとした上で、前記データベースのレコードの更新処理を実行するようにしたものである。

【0019】請求項1乃至請求項3記載の発明により、同一レコードに対する複数の更新処理を単一のアクセスにより一回で処理することができるため、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現できる。

【0020】請求項4記載の発明は、請求項1乃至請求項3記載のデータベース更新方法において、マージ処理により生成される新更新レコードを順次保存する一方、マージ処理の対象としたその更新レコードと同一IDを有する受付済の更新レコードを順次消去するようにしたものである。これにより、マージ処理実行の度に不要となった旧レコードが消去され、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【0021】請求項5記載の発明は、請求項1乃至請求項4記載のデータベース更新方法において、最終的に削除対象となるレコードと同一レコードに対する登録指示又は更新指示がある場合には、その登録指示又は更新指示を無効にするようにしたものである。これにより、後

に削除されるレコードについての無駄な登録処理や更新処理を実行する必要がなくなるため、バッチ処理、データベースのレコードの更新処理自体の迅速化が実現できる。

【0022】請求項6記載の発明は、請求項1乃至請求項4記載のデータベース更新方法において、データベースのレコードの更新処理実行前の更新レコードを登録バッチファイルと更新バッチファイルと削除バッチファイルとに分割して管理するようにしたものである。これにより、予め処理ファイルをグルーピングして整列するため、ファイル管理が容易になり、バッチ処理、データベースのレコードの更新処理自体を簡易、迅速に実行できる。

【0023】請求項7記載の発明は、請求項6記載のデータベース更新方法において、登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一IDのレコードをデータベースのレコード更新以前に消去するようにしたものである。これにより、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【0024】請求項8記載の発明は、請求項1乃至請求項7記載のデータベース更新方法において、レコードの更新指示を時刻管理し、同一レコードに対して複数の更新指示がある場合は、新しい時刻の更新指示のみを実行するようにしたものである。これにより、同一レコードに再度更新がかけられた結果不要となった旧更新処理をバッチ処理対象から削除できるため、バッチレコード数を削減できバッチ処理の負荷を軽減させることができる。

【0025】請求項9記載の発明は、請求項1乃至請求項7記載のデータベース更新方法において、データベースに対する検索の実行の後に、その検索結果レコードと更新処理待ちの更新レコードとをマージ処理するようにしたものである。これにより、データベースのレコードの実体に対するバッチ処理に時間がかかり更新が遅れた場合であっても、クライアントが実行する検索の結果は、常に、更新後の最新のものとなる。

【0026】請求項10記載の発明は、請求項1乃至請求項7記載のデータベース更新方法において、データベースに登録されたレコードの更新指示受け付けの際にレコードの更新に要する時間を計算し、その時間が一定値以下である場合には、更新処理をリアルタイムで実行し、その時間が一定値以上である場合には、更新処理をバッチ処理で実行するようにしたものである。これにより、短時間で更新処理が完了する場合には即時に更新処理を実行し、更新処理に時間がかかる場合には、バッチ処理により更新処理を実行するため、クライアント、サーバ型データベースシステム全体として最適の運用が可能になる。

【0027】以下、本発明の実施の形態について、図面を用いて、具体的に説明する。図1は、本発明のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図である。図1において、1は、WS、PC、ホストマシン等のサーバマシン、2は、検索、登録、更新、削除処理を要求するアプリケーションを搭載した、WS、PC等のクライアントマシンであり、サーバマシン1とクライアントマシン2とは、LAN、WAN等のネットワーク3で接続されている。4は、サーバマシン1に実装されたデータベース処理装置、5は、クライアントからの処理要求を受取り、オンライン時にクライアントからの処理要求を受取り、検索、登録、更新、削除処理を実行するデータベース処理部、20は、クライアントマシン2からのレコードの登録、更新、削除の要求に応じ、オンライン時にリアルタイムにバッチファイルを作成するバッチファイル管理部、21は、オフライン時にバッチファイルのレコードのバッチ処理を実行するバッチ処理部22は、オンライン時にリアルタイムにデータベース10を構成するレコードの検索処理等を実行するオンライン処理部である。また、23は、登録バッチファイル24への登録レコードの追加、更新、削除処理を行う登録バッチファイルアクセス部、25は、更新バッチファイル26への更新レコードの追加、更新、削除処理を行う更新バッチファイルアクセス部、27は、削除バッチファイル28への削除レコードの追加、更新、削除処理を行う削除バッチファイルアクセス部である。

【0028】以上のように構成されたクライアント、サーバ型データベースシステムデータベース処理装置におけるデータベース更新方法について、具体的に、図面を用いて説明する。図2は、本発明のデータベース更新方法の全体処理フロー図である。

【0029】まず、クライアントマシン2のアプリケーションは、サーバマシン1のデータベース処理装置4に対し、登録、更新、削除レコードを送信し、登録処理、更新処理、削除処理、検索処理を依頼する(ST201)。データベース処理部5は、登録、更新、削除レコードを受取り、登録処理、更新処理、削除処理、検索処理要求を受け付けて(ST202)、依頼内容に応じて異なる制御を実行する。

【0030】検索処理の場合には、データベース処理部5は、オンライン処理部22を起動し(ST203)、検索を実行し、データベースから検索条件に合致したレコードを取得し、クライアントマシン2のアプリケーションへ検索結果を返す(ST204)。

【0031】登録処理の場合には、データベース処理部5は、バッチファイル管理部20を起動し(ST205)、バッチファイル管理部20が、登録バッチファイルアクセス部23により登録バッチファイル24に登録レコードを追加する(ST206)。

【0032】次に、更新処理について説明する。更新処理の場合には、データベース処理部5は、図3の更新処理1の処理フロー図に従って、更新処理を実行する。まず、バッチファイル管理部20が起動され(ST301)、更新処理の対象となっているレコードと同一IDのレコードが新規登録の対象となっていないかをチェックする。具体的には、登録バッチファイルアクセス部23により更新レコードと同一IDのレコードが登録バッチファイル24にあるか否かチェックする(ST302)。

【0033】登録バッチファイル24内に同一レコードがある場合には、登録バッチファイルアクセス部23により、更新レコードと登録バッチファイル24内の既存の登録レコードとをフィールド単位でマージ処理し新規登録レコードを生成した後、既存の登録レコードを削除し、生成した新規登録レコードを削除した既存の登録レコードと置換し新たに追加する(ST303)。例えば、図4に示すように、レコードIDが100のレコードのフィールド1を”さささ”に更新する更新レコード29をクライアントのアプリケーションから受け取った場合、登録バッチファイルに既に登録されている登録レコード30は、フィールド1が”あああ”から”さささ”に訂正され新規登録レコード31として登録バッチファイルに記憶される。

【0034】登録バッチファイル24内に同一レコードがない場合には、次に、バッチファイル管理部20は、更新バッチファイルアクセス部25により、更新レコードと同一IDのレコードが更新バッチファイル26にあるか否かをチェックする(ST304)。

【0035】同一IDのレコードが更新バッチファイル26にある場合には、バッチファイル管理部20は、更新バッチファイルアクセス部25により、更新レコードと更新バッチファイル26内の既存の更新レコードとをフィールド単位でマージ処理し新規更新レコードを生成した後、既存の更新レコードを削除し、生成した新規更新レコードを削除した更新レコードと置換する(ST305)。例えば、図5に示すように、レコードIDが100のレコードのフィールド1を”さささ”に更新する更新レコード32をクライアントのアプリケーションから受け取った場合、更新バッチファイル26に既に登録されているレコードIDが100のレコードのフィールド3を”すすす”に更新する更新レコード33は、フィールド1を”さささ”に、フィールド3を”すすす”に更新する更新レコード34に更新される。

【0036】同一IDのレコードが更新バッチファイル25にない場合には、次に、バッチファイル管理部20は、削除バッチファイルアクセス部27により、更新レコードと同一IDのレコードが削除バッチファイル28にあるか否かをチェックする(ST306)。同一IDのレコードが削除バッチファイル28にある場合には、

その更新レコードは、更新不要のレコードであるから、バッチファイル管理部20は、受付けた更新レコードを破棄する(ST307)。

【0037】同一IDのレコードが削除バッチファイル28にない場合に、初めて、バッチファイル管理部20は、更新バッチファイルアクセス部25により、更新バッチファイル26へ受付けた更新レコードを追加する(ST308)。

【0038】このようにして、更新レコードと同一IDのレコードが登録バッチファイル及び削除バッチファイルに存在するかどうかをチェックし、既に登録バッチファイルか更新バッチファイルに同レコードが存在する場合には、双方のレコードをフィールド単位でマージ処理したレコードで既存のバッチファイル中のレコードを更新する処理を、更新レコードを受付ける度に実行することにより、同一のレコードに対する登録処理、更新処理を、常に、単一のレコードを基準として管理し処理できるため、バッチ処理対象となる実レコード数の飛躍的削減が可能となる。また、既に削除バッチファイルに同一IDのレコードが保持されていることが確認できた場合には、同一IDの登録レコード、更新レコード全てが消去されるため、バッチレコード数を大幅な削減が可能となる。

【0039】次に、削除処理について説明する。削除処理の場合には、データベース処理部は、図6の削除処理1のフロー図に従って、削除処理を実行する。まず、データベース処理部5は、バッチファイル管理部20を起動する(ST601)。

【0040】バッチファイル管理部20は、登録バッチファイルアクセス部23により削除レコードと同一IDのレコードが登録バッチファイル24にあるか否かをチェックする(ST602)。ある場合は、バッチファイル管理部20は、登録バッチファイル24からそのレコードを削除する(ST603)。

【0041】無い場合は、次に、バッチファイル管理部20は、削除レコードと同一IDのレコードが更新バッチファイル26にあるか否かをチェックする(ST604)。ある場合には、バッチファイル管理部20は、更新バッチファイル26からそのレコードを削除する(ST605)。

【0042】無い場合には、次に、バッチファイル管理部20は、削除レコードと同じレコードが削除バッチファイル119にあるか否かをチェックする(ST606)。ある場合には、その削除レコードは、重複していることになり再登録の必要はないため、バッチファイル管理部20は、受付けた削除レコードを破棄する(ST607)。

【0043】同一IDのレコードが削除バッチファイル28にない場合、バッチファイル管理部20は、初めて、削除バッチファイル28に受付けた削除レコードを

追加する (ST608)。

【0044】このようにして、登録バッチファイル又は更新バッチファイルに存在する削除レコードと同一IDのレコードは、その登録処理、更新処理自体が必要ないわけであるから、これら全てを消去することにより、バッチレコード数が削減されることとなる。例えば、図7のように更新処理を時系列に受付けた場合、従来のデータベース処理装置におけるバッチ処理では合計8レコード全てについての処理が必要となるが、本発明によれば、更新等の処理を受付ける度に、同一IDのレコードの検索、マージ処理の実行をしてバッチファイルの集約処理を行う結果、バッチ処理の対象レコード数は、図8に示すように、合計4レコードになる。

【0045】以上の処理により、データベース10に対する登録、更新、削除待ちのバッチファイルは、各レコードIDにつき単一のレコードに集約されるため、本発明では、更に、これをクライアントマシン2からの検索依頼があった場合に、その検索結果に反映させることとした。

【0046】この処理は、オンライン処理部22の機能の一部として実行される。図9の検索処理フロー図を用いて具体的に説明する。まず、クライアントマシン2アプリケーションから検索処理要求を受け取った場合、データベース処理部5は、オンライン処理部22を起動し (ST901)、検索を実行し、データベース10から検索条件に合致したレコードを取得し保持する (ST902)。

【0047】次いで、オンライン処理部22は、検索結果のレコードの集合から、1レコードを取り出し (ST904)、バッチファイル管理部20を起動して (ST905)、各バッチファイルに同一IDのレコードがあるか否かをチェックする。

【0048】バッチファイル管理部20は、先ず、削除バッチファイルアクセス部27により、削除バッチファイル28に同一IDのレコードがあるかチェックさせる (ST906)。ある場合は、これをオンライン処理部22に通知し (ST907)、オンライン処理部22により、検索結果レコードから該当レコードを削除する (ST908)。

【0049】削除バッチファイル28に同一IDのレコードが無い場合は、次に、バッチファイル管理部20は、更新バッチファイルアクセス部25により、更新バッチファイル26に同一IDのレコードがあるか否かをチェックする (ST909)。

【0050】更新バッチファイル26に同一IDのレコードがある場合は、更新バッチファイルアクセス部25は、更新バッチファイル26の更新レコードが検索条件に合致するかチェックし (ST910)、合致する場合には、同一IDのレコードが更新バッチファイル26に存在することを、オンライン処理部22に通知し (ST

911)、オンライン処理部22は検索結果レコードと更新バッチファイル26の更新レコードとのマージ処理を実行し、新たに生成された新規更新レコードで検索結果レコードを更新する (ST912)。合致しない場合には、それをオンライン処理部22に通知して、検索結果レコードを破棄する (ST913)。これは、バッチ処理実行前であるためデータベース10の実レコードは更新されていないものの、既にいずれかのクライアント2からのレコード更新要求が受け付けられており、現在の検索結果レコードは、本来検索条件には、合致しないレコードであるからである。

【0051】更新バッチファイル26に同一IDのレコードが無い場合は、次の検索結果レコードについて同様の処理を実行する (ST914)。

【0052】このように、ST904からST914までの処理を、データベースから検索した検索結果レコードがなくなるまで繰返して実行する (ST903)。

【0053】以上の削除バッチファイル28と更新バッチファイル26との処理ループを抜けた後、オンライン処理部22は、登録バッチファイルアクセス部23により、登録バッチファイル24の登録レコードに対し検索を実行し、検索条件に合致した登録レコードを取得し (ST915)、オンライン処理部22は、該当レコードを検索結果レコードに追加する (ST916)。

【0054】以上の検索処理が実行された後、データベース処理部5は、クライアントのアプリケーションに対して検索結果レコードを送信する。

【0055】以上の処理により、データベース10の実レコードの更新前であっても、結果的に検索対象とされるため、レコードの更新にバッチ処理を採用しつつも、クライアントに対しては最新の検索結果を提供することができ、検索のリアルタイム性を確保できる。

【0056】次に、本発明の第二の実施の態様を説明する。本実施態様では、更に、バッチ処理部21による、データベース10に対する各バッチファイルの更新処理の実行タイミングを、システム全体のパフォーマンスを考慮して調整しており、データベース10の更新処理が短時間で完了する場合はオンラインにより実行し、時間を要する場合はバッチ処理により実行することとしている。図10および図11の処理フロー図を用いて具体的に説明する。

【0057】この処理も、オンライン処理部22の機能の一部として実行される。まず、クライアントマシン2のアプリケーションは、データベース処理装置4に対し、登録処理、更新処理、削除処理、検索処理を依頼し (ST1001)、データベース処理部5は、これを受け付け (ST1002)、検索処理であればこれを実行する (ST1003)。

【0058】登録処理の場合には、データベース処理部5は、CPU負荷状況、データベース内レコード数を取

得し、それらの値より現負荷時の登録処理時間を計算し、規定時間で処理可能かをチェックする(ST1004)。その際、例えば、登録処理に対する規定時間、負荷なし時リアルタイム処理時間[対単位レコード長]、処理時間増加比率[対CPU負荷]、処理時間増加比率[対単位DBレコード数]はあらかじめ図12の処理時間テーブルのように設定し、その計算式は図13のように設定する。

【0059】規定時間で済む場合には、データベース処理部5は、オンライン処理部22により、データベース10に対しリアルタイムに登録処理を実行し(ST1005)、規定時間を超える場合には、通常の場合と同様に登録バッチファイルへ登録処理を行う(ST1006からST1007)。

【0060】例えば、登録レコード長が200バイト、OSから取得したサーバマシンのCPU負荷が30%、データベースレコード数が100レコードとすると、現負荷時の1レコードあたりのリアルタイム登録処理時間は、6.8秒となる(図12の設定で図13の計算式による)。仮に、あらかじめ設定された登録処理に対する規定時間が7秒だとすると、規定時間で処理可能なため、この登録処理は、データベース10へリアルタイムに登録される。

【0061】更新処理の場合には、データベース処理部810は、CPU負荷状況、データベース内レコード数、を取得しそれらの値より現負荷時の更新処理時間を計算し、現負荷時のリアルタイム更新処理時間の規定時間で処理可能かをチェックする(ST1010)。規定時間で済む場合には、データベース処理部5は、バッチファイル管理部20により、登録、更新、削除バッチファイルに同レコードが存在するかチェックする(ST1011)。存在する場合には、そのままオンラインでデータベース10のレコードを更新すると各バッチファイルのレコードと不整合が発生するため、既に説明したマージ処理を含む各バッチファイルに対する更新処理を実行する(ST1012)。存在しない場合にはリアルタイムに更新処理を実行する(ST1013)。

【0062】規定時間を超える場合には、通常の更新と同様に各バッチファイルに対する更新処理を実行する(ST1014)。

【0063】削除処理の場合にも、同様に、削除処理時間を計算し、規定時間で処理可能かをチェックする(ST1015)。規定時間で済む場合は、バッチファイル管理部20により、登録、更新、削除バッチファイルに同レコードが存在するかチェックし(ST1016)、存在する場合は各バッチファイルに対する削除処理を実行し(ST1017)、存在しない場合はリアルタイムに削除処理を実行する(ST1018)。規定時間を超える場合は、各バッチファイルに対する削除処理を実行する(ST1019)。

【0064】以上のように、レコードのデータ量やサーバマシンの負荷が少なく、データベースへのリアルタイムでの登録、更新、削除処理が短時間で済む場合、リアルタイムでデータベースノレコードに対する処理を実行することにより、オフライン時のバッチ処理の負荷を減少させることができる。

【0065】

【発明の効果】本発明により、同一レコードに対する複数の更新処理を単一のアクセスにより一回で処理することができるため、バッチ処理、データベースのレコードの更新処理自体の迅速化を実現できる。

【0066】また、マージ処理実行の度に不要となった旧レコードが消去され、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【0067】また、後に削除されるレコードについての無駄な登録処理や更新処理を実行する必要がなくなるため、バッチ処理、データベースのレコードの更新処理自体の迅速化が実現できる。

【0068】また、予め処理ファイルをグルーピングして整列するため、ファイル管理が容易になり、バッチ処理、データベースのレコードの更新処理自体を簡易、迅速に実行できる。

【0069】また、更新レコードを保持するメモリの有効活用ができ、バッチ処理、データベースのレコードの更新処理自体の迅速化も実現できる。

【0070】また、同一レコードに再度更新がかけられた結果不必要となった旧更新処理をバッチ処理対象から削除できるため、バッチレコード数を削減できバッチ処理の負荷を軽減させることができる。

【0071】また、データベースのレコードの実体に対するバッチ処理に時間がかかり更新が遅れた場合であっても、クライアントが実行する検索の結果は、常に、更新後の最新ものとなる。

【0072】また、短時間で更新処理が完了する場合には即時に更新処理を実行し、更新処理に時間がかかる場合には、バッチ処理により更新処理を実行するため、クライアント、サーバ型データベースシステム全体として最適の運用が可能になる。

40 【図面の簡単な説明】

【図1】本発明のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図

【図2】本発明のデータベース更新方法の全体処理フロー図

【図3】本発明のデータベース更新方法の更新処理フロー図

【図4】本発明の更新バッチファイル内のレコードを表す図

【図5】本発明の更新バッチファイル内のレコードを表す図

13

【図6】本発明のデータベース更新方法の削除処理フロー図

【図7】本発明の時系列更新依頼レコードを表す図

【図8】本発明の各バッチファイル内のレコードを表す図

【図9】本発明のデータベース更新方法の検索処理フロー図

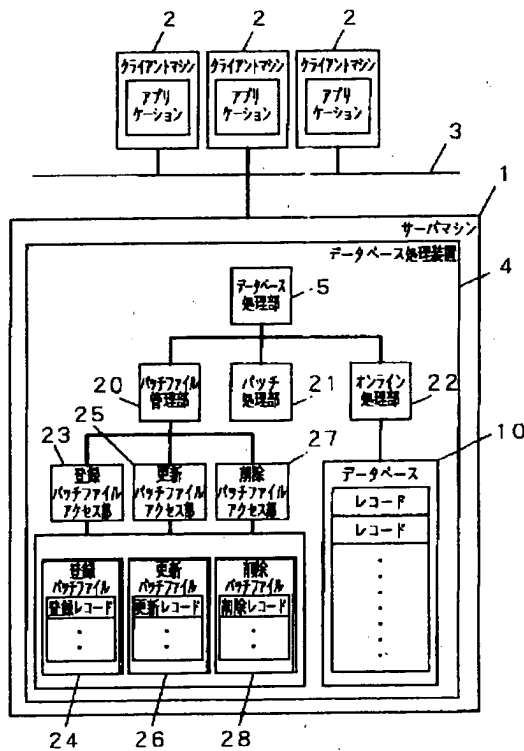
【図10】本発明の第2実施態様によるデータベース更新方法の処理フロー図

【図11】本発明の第2実施態様によるデータベース更新方法の処理（続き）フロー図

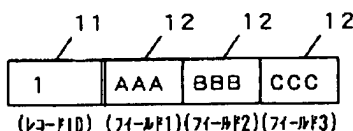
【図12】本発明の第2実施態様によるデータベース更新方法に使用する処理時間テーブルを表す図

【図13】本発明の第2実施態様によるデータベース更新方法に使用する処理時間の計算方法を示す図

【図1】



【図15】



14

【図14】従来のデータベース更新方法を実現するクライアント、サーバ型データベースシステムの全体構成図

【図15】従来のデータベース内のレコードを表す図

【図16】従来のデータベース更新方法の全体処理フロー図

【図17】従来のバッチファイル内のレコードを表す図

【符号の説明】

- 10 データベース
- 20 バッチファイル管理部
- 21 バッチ処理部
- 22 オンライン処理部
- 24 登録バッチファイル
- 26 更新バッチファイル
- 28 削除バッチファイル

【図4】

更新レコード

(処理種別) (レコードID) (フィールド1) (フィールド2) (フィールド3)				
更新	100	さささ	-	-

↓

バッチファイル内レコード（更新前）

登録	100	あああ	いいい	ううう
----	-----	-----	-----	-----

↓

バッチファイル内レコード（更新後）

登録	100	さささ	いいい	ううう
----	-----	-----	-----	-----

【図5】

更新レコード

(処理種別) (レコードID) (フィールド1) (フィールド2) (フィールド3)				
更新	100	さささ	-	-

↓

更新バッチファイル内レコード（更新前）

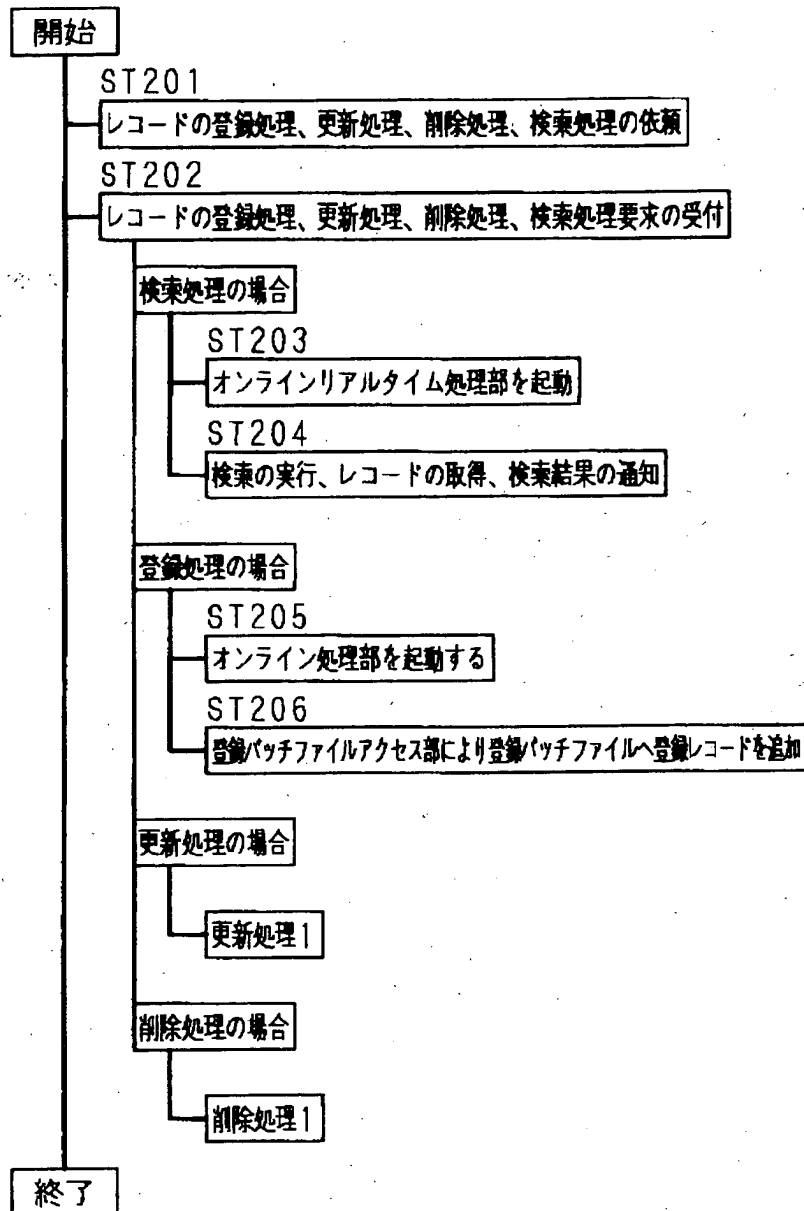
更新	100	-	-	すすす
----	-----	---	---	-----

↓

更新バッチファイル内レコード（更新後）

更新	100	さささ	-	すすす
----	-----	-----	---	-----

【図2】



【図12】

処理	規定時間	負荷なし時リアルタイム処理時間 【対単位レコード量】	処理時間増加比率 【対CPU負荷】	処理時間増加比率 【対単位DBレコード量】
登録	7秒	1秒 / 100バイト	3% / 1%	3% / 1件
更新	8秒	0.5秒 / 100バイト	5% / 1%	5% / 1件
削除	12秒	1.5秒 (固定)	7% / 1%	7% / 1件

【図8】

登録バッチファイル内レコード

(処理種別) (レコードID) (フィールド1) (フィールド2) (フィールド3)

登録	100	aaa	iii	uuu
----	-----	-----	-----	-----

登録	101	caa	kkk	lll
----	-----	-----	-----	-----

登録	102	hhh	lll	jjj
----	-----	-----	-----	-----

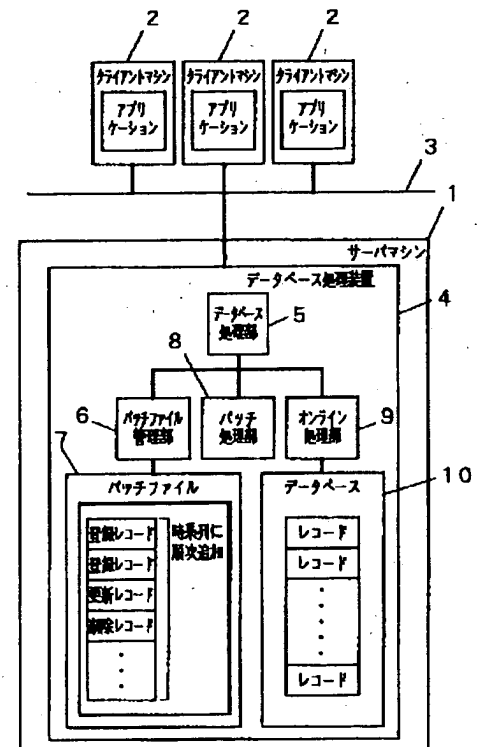
更新バッチファイル内レコード

レコードなし

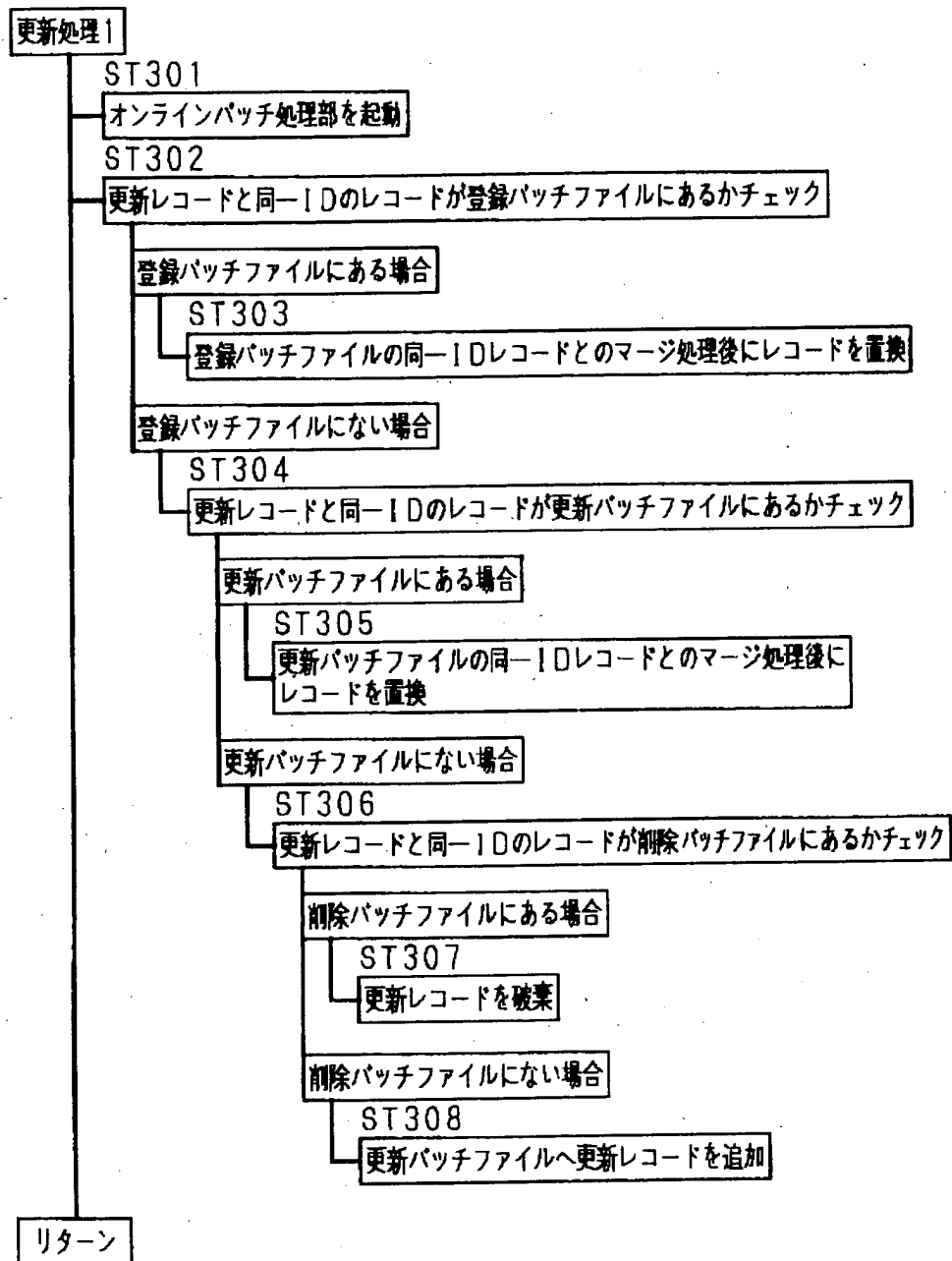
削除バッチファイル内レコード

削除	1	-	-	-
----	---	---	---	---

【図14】



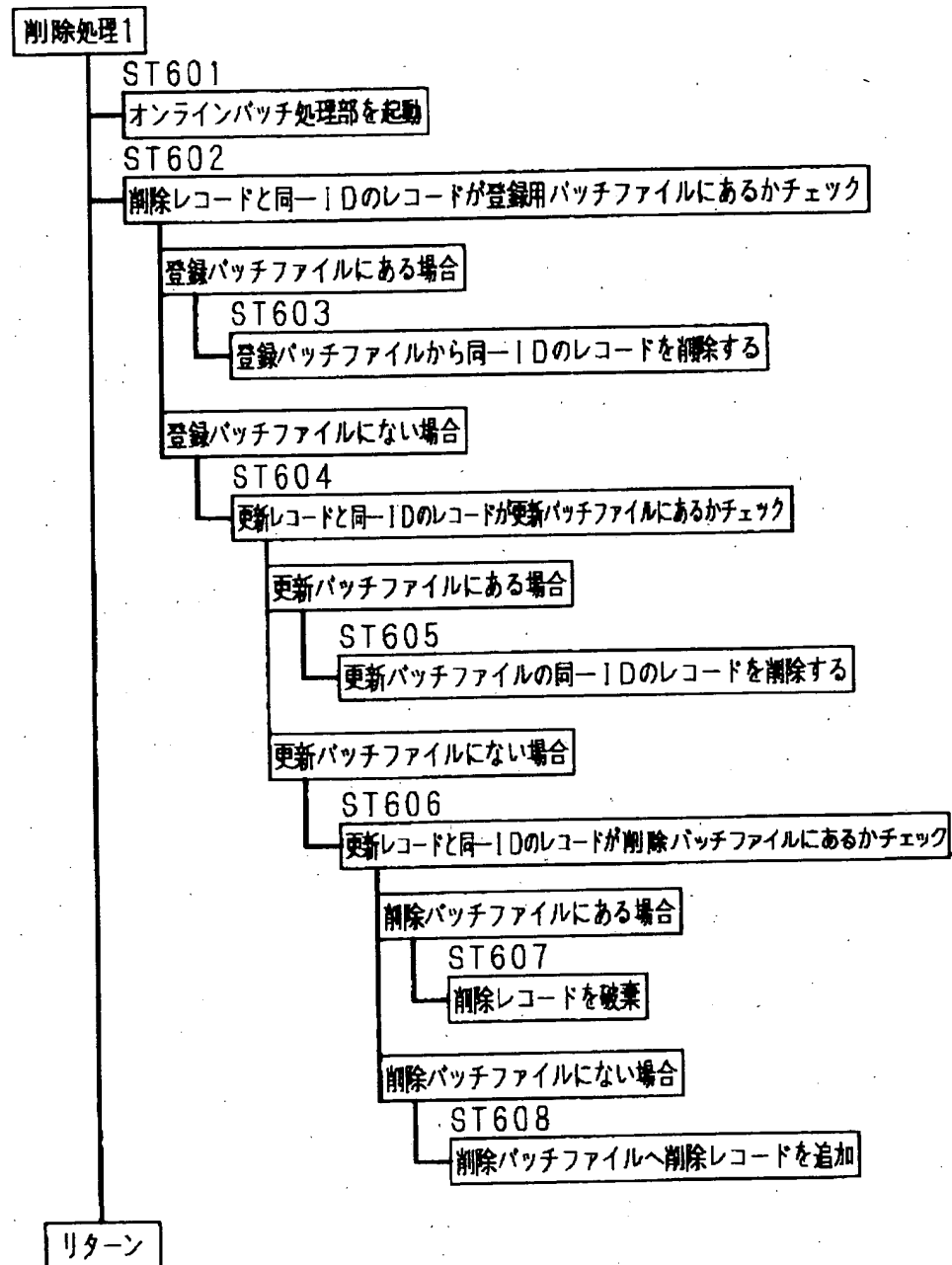
【図 3】



【図 1 3】

$$\begin{aligned}
 \text{現負荷時リアルタイム処理時間} &= \{ \text{負荷なし時リアルタイム処理時間 [対単位レコード長]} \\
 &+ (\text{負荷なし時リアルタイム処理時間 [対単位レコード長]} * \text{CPU負荷状況} * \text{処理時間増加比率 [対CPU負荷]}) \\
 &+ (\text{負荷なし時リアルタイム処理時間 [対単位レコード長]} * \text{DBレコード件数} * \text{処理時間増加比率 [対単位DBレコード数]}) \} \\
 &* (\text{レコード長} / \text{単位レコード長})
 \end{aligned}$$

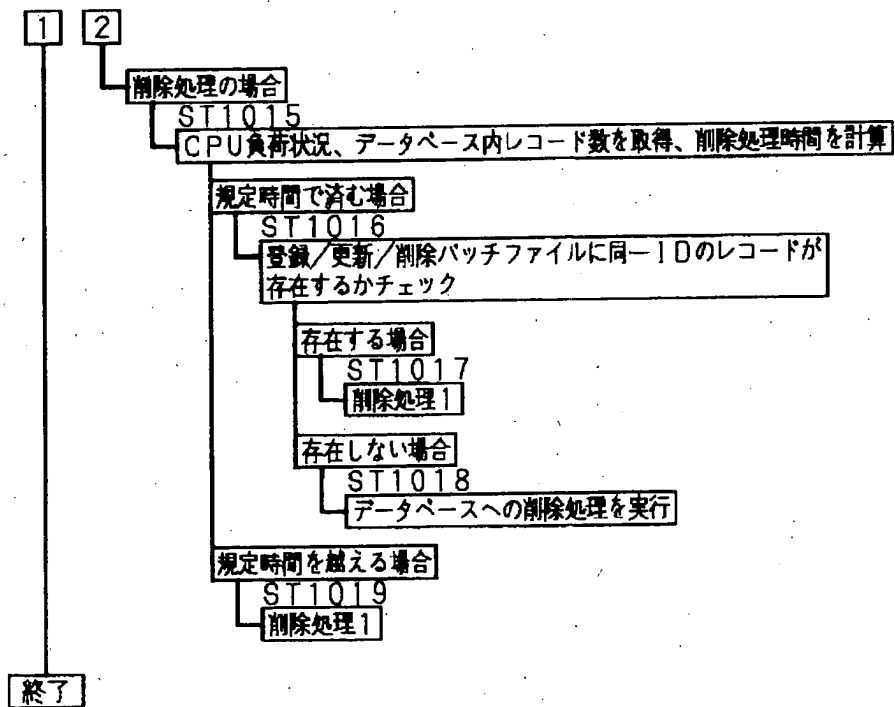
【図6】



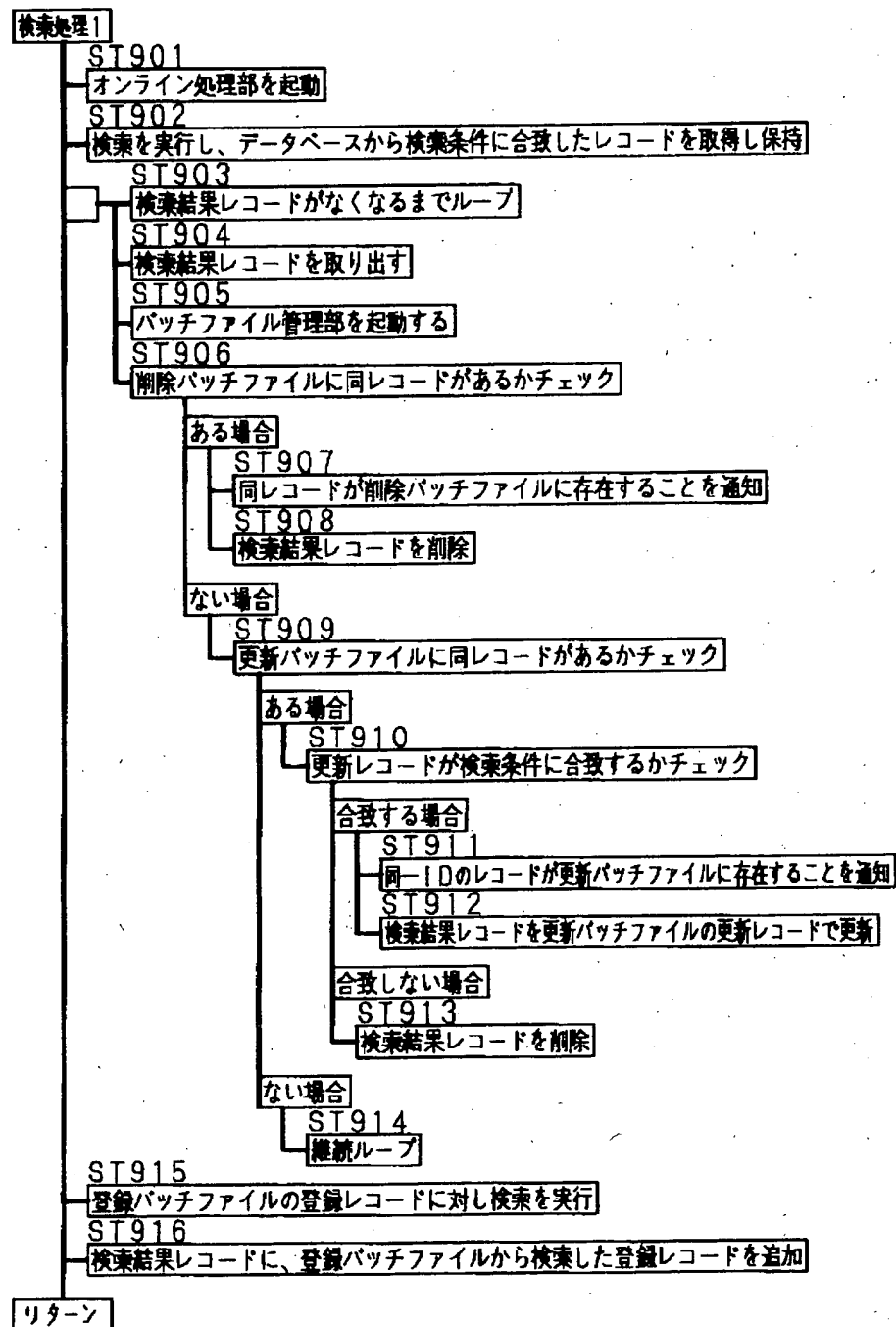
【図 7】

(処理種別) (レコードID) (フィールド1) (フィールド2) (フィールド3)					
登録	100	あああ	いいい	ううう10:00
登録	101	かかか	ききき	くくく10:10
更新	1	EEE	-	-10:20
更新	1	-	FFF	-10:30
更新	1	-	-	GGG10:40
更新	1	さささ	-	-10:50
削除	1	-	-	-11:00
登録	102	HHH	III	JJJ11:10

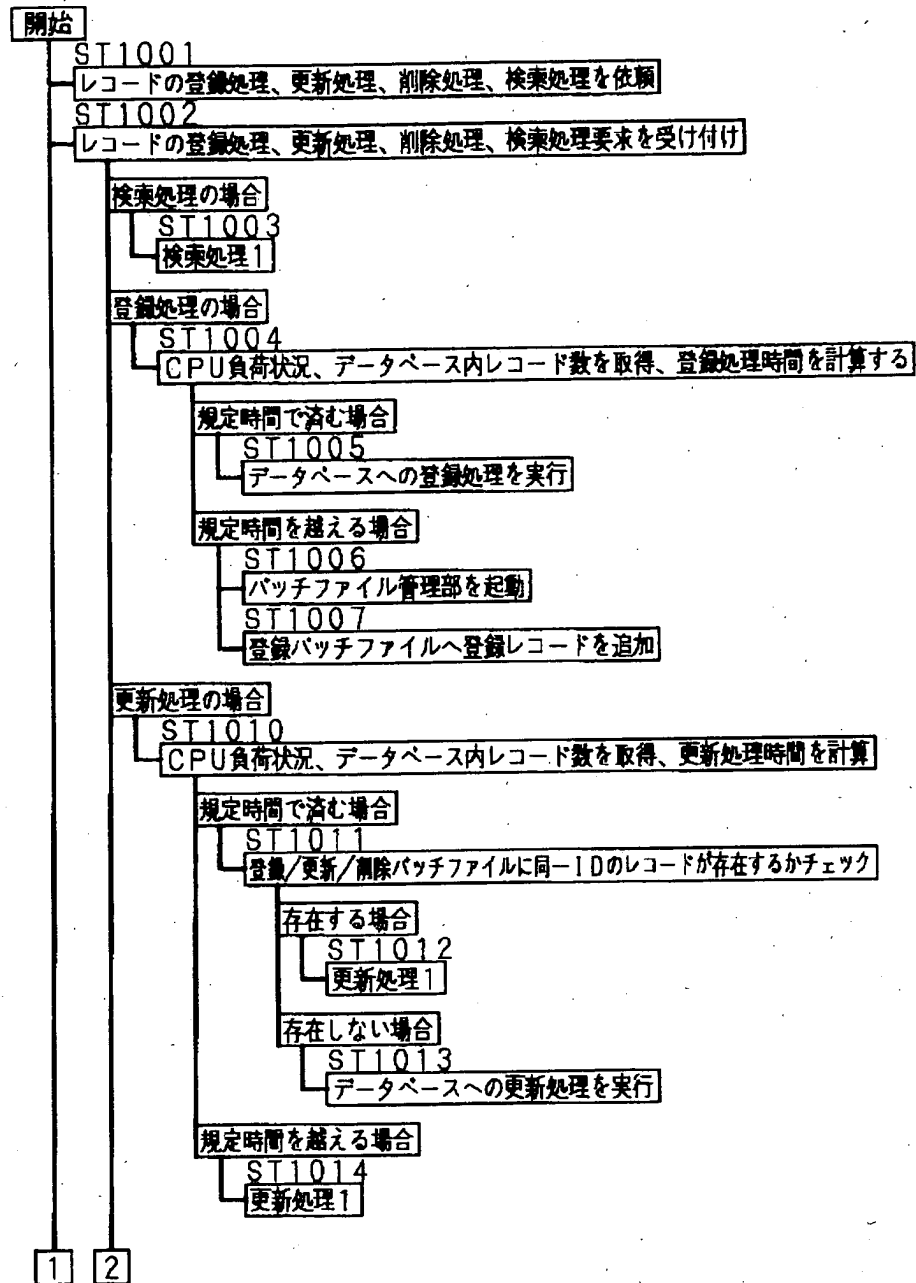
【図 11】



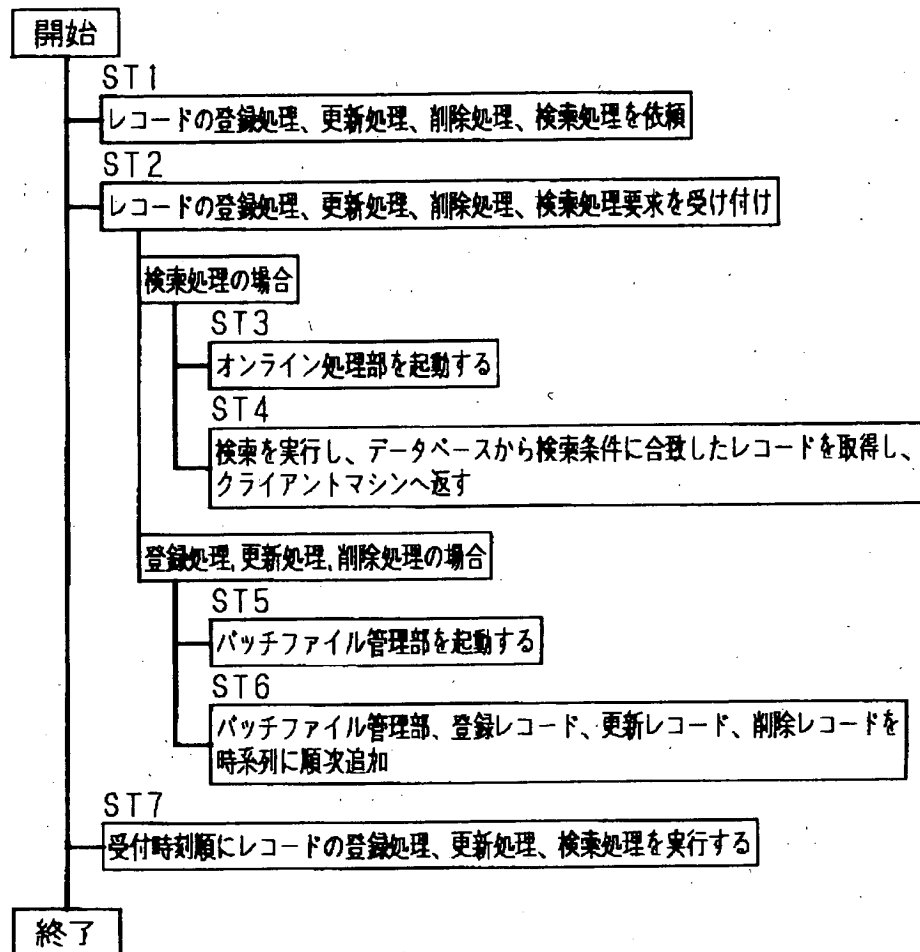
【図 9】



【図10】



【図 16】



【図 17】

	13 (処理種別)	11 (レコードID)	12 (ファイル1)	12 (ファイル2)	12 (ファイル3)	
登録	100	あああ	いいい	ううう	10:00
登録	101	かかか	ききき	くくく	10:10
更新	1	EEE	—	—	10:20
更新	1	—	FFF	—	10:30
更新	1	—	—	GGG	10:40
削除	1	—	—	—	10:50
更新	1	さささ	—	—	11:00
登録	102	HHH	III	JJJ	11:10